EURASIP Journal on Applied Signal Processing 2003:2, 199–208

Motion Pattern-Based Video Classification and Retrieval

Yu-Fei Ma
Microsoft Research Asia, Beijing Sigma Center, No. 49, Zhichun Road, Hai Dian District, Beijing 100080, China
Email: yfma@microsoft.com

Hong-Jiang Zhang
Microsoft Research Asia, Beijing Sigma Center, No. 49, Zhichun Road, Hai Dian District, Beijing 100080, China
Email: hjzhang@microsoft.com

Received 25 March 2002 and in revised form 3 November 2002

Today's content-based video retrieval technologies are still far from human's requirements. A fundamental reason is the lack of content representation that is able to bridge the gap between visual features and semantic conception in video. In this paper, we propose a motion pattern descriptor, *motion texture* that characterizes motion in a generic way. With this representation, we design a semantic classification scheme to effectively map video clips to semantic categories. Support vector machines (SVMs) are used as the classifiers. In addition, this scheme also improves significantly the performance of motion-based shot retrieval due to the comprehensiveness and effectiveness of motion pattern descriptor and the semantic classification capability as shown by experimental evaluations.

Keywords and phrases: motion pattern descriptor, video classification, video retrieval, machine learning.

1. INTRODUCTION

The management and access of a mass volume of multimedia data, video in particular, is an entry barrier for better user's experiences. Content-based video retrieval has been proposed as a solution to address this problem. However, the success is very limited. One of the important barriers is the lack of comprehensive, compact, and flexible representation of video content. Current content-based technologies depend mostly on low-level features, which are extracted fully automatically, but bear little or no semantic content of video. It is understood that semantic representation and classification are the foundations for building an effective and efficient index of video data. However, when the textual information is not available or impossible to be extracted, we have to resort to low-level features. Then, the challenge is how to bridge the gap between low-level feature and semantic conception. In other words, we need to develop a comprehensive and effective video content representation that is able to bridge the gap between visual features and semantic conception in video.

In this paper, we present our work on the extraction and application of motion feature which is the most distinctive character of video. We propose a motion pattern descriptor, *motion texture*, to efficiently characterize the motion features of video in a generic way. With this motion representation, a semantic classification scheme is designed to map motion texture to the semantic conceptions by kernel support vector machines (SVMs) method. In addition, we present a method to take advantage of the proposed semantic classification to enhance the performance of traditional content-based video retrieval.

The rest of the paper is organized as follows. Related works are reviewed in Section 2. Section 3 introduces the proposed motion pattern descriptor, *motion texture*, in detail. Then, a motion pattern-based semantic classification scheme is presented in Section 4. Also, the kernel support machines are reviewed briefly in this section. In Section 5, we present the application scheme and experimental results of motion texture and semantic classification scheme in shot retrieval to enhance the performance of motion-based retrieval. Section 6 concludes the paper.

2. RELATED WORKS

As an important cue in understanding video content, motion has been a study topic ever since computer vision and image processing research started. Motion estimation is a conventional method to extract motion information from two consecutive frames [1]. Parametric global motion estimation generates the parametric model of camera motion or

dominant motion such as affine model. Nonparametric motion estimation generates a field of displacement pairs such as optical flow. Both of the two results can be looked upon as motion representation. Recently, many approaches have been proposed to apply motion analysis to content-based video retrieval. For example, optical flow is used for video indexing in [2]. However, with such motion representations, motion feature cannot be utilized efficiently and sufficiently because parametric global motion descriptors are too coarse, and nonparametric motion descriptors are often over fine. A reasonable representation of motion is trajectories of moving objects since humans usually only concerned with object motion, and do not perceive the existence of camera motion. In order to extract motion trajectories, object segmentation and tracking [3], or motion layer extraction [4], is often adopted. However, the exact object trajectories usually are unavailable due to the unreliability of automatic objects extraction. Therefore, as a simplified alternative, trajectories of moving regions are extracted to facilitate video retrieval in [5]. Another motion extraction method is based on the temporal slices of image volume [6, 7, 8]. In [8], Ngo characterized motion using multiple slices and tensor measurement. Such temporal slices encode rich motion clues which are very useful for specific motion characterization, but they also have many confusable visual patterns that cannot be suppressed easily.

While motion patterns often convey some semantic information, especially in sports video, they have been used often only as a low-level attribute in most of the existing video analysis systems. By extracting texture and motion features, the scene contents of video are classified into 9 categories in [9]. The categories are defined according to 3 levels of texture complexity and 3 levels of motion intensity each, namely, low, medium, and high. In [10], motion histogram, dominant motion, and model parameters of global motion estimation are extracted from each frame. By integrating with color and audio features, the shots of TV programs are classified into 5 categories: news reports, weather forecasts, commercials, basketball games, and football games. The work [11] focuses on basketball events classification in which motion, color, and edge features are used to classify basketball events. Motion features include the orientation of dominant motion and average magnitude of motion vectors during a video clip. All of these statistical motion descriptors represent most of global motion information, but they cannot represent the temporal variation pattern in video clip sufficiently. In addition, the categories defined in these literatures are only based on non-semantic measurements or a limited scope of semantic conceptions. Lacking of a generic classification scheme is a main constraint of these methods.

Another key issue in classification is classifier selection. Some systems took advantage of HMM-based methods, such as [10]. An entropy-based inductive tree-learning algorithm was used in [11]. Neural networks are also good choices for classifiers. Radial basis function network, feed forward network, recurrent network, and so on were all often adopted for this purpose [12]. In recent years, machine learning was successfully applied to multimedia classification such as SVM [13]. Kernel SVMs is a good optimal classifier due to its high generalization performance without the need to add a priori knowledge, even when the dimension of the input space is very high.

It is difficult to use motion information effectively in video retrieval since the motion information is always hidden behind temporal variances of other visual features such as color, shape, and texture. It is necessary to extract motion information from original image sequence and put it into an explicit format of motion representation. Besides the motion representations proposed in [5, 6, 7, 8], motion vector field (MVF) in MPEG stream is also used for fast video indexing or retrieval such as in [14, 15]. However, these motion representations were only used as low-level features in retrieval applications. So the retrieval results were far from human's requirements at the semantic level.

In summary, inspite of many research efforts, content-based video analysis and retrieval are far from being an effective solution due to two main constraints: the lack of efficient content representation and the lack of an effective method for bridging the gap between low-level features and semantic conceptions. Removing these two constraints are the objectives of the work presented in this paper.

3. MOTION PATTERN DESCRIPTOR

As reviewed in Section 2, parametric global motion estimation generates the parametric model of camera motion or dominant motion such as affine model; nonparametric motion estimation generates a field of displacement pairs such as optical flow. The result of nonparametric motion estimation is also often referred to as MVF. In this paper, we propose a method that takes advantage of the results of nonparametric motion estimation to generate a motion pattern descriptor.

MVF can be a field of optical flow obtained by pixel-wised motion estimation. It also can be a sparser field generated by block-based motion estimation such as in MPEG encoding process. Though real motion cannot be obtained by block-based motion-estimation algorithm (BMA), the lost is light for video-content analysis. In this paper, we adopt the MVF in MPEG stream as approximate block-based motion estimation to create a motion pattern descriptor, named *motion texture*.

The motion texture is extracted from MVFs by three steps. First, we transform MVF to an energy unit circle (EUC) by circular mapping. Then, the consecutive EUCs are transformed to a texture image by slicing called directional slices. Finally, the directional slices are measured by moments to form a multidimensional vector. Such a multidimensional vector describes the motion pattern in a compact way and is used as motion pattern descriptor, the *motion texture*.

3.1. Circular mapping

In a given MVF, let (i, j) be the position of macro blocks in the raster-scan order, and $V_{i,j}(\Delta x_{i,j}, \Delta y_{i,j})$ be the motion vector of macro block $\mathrm{MB}_{i,j}$, then we define the energy in macro block $\mathrm{MB}_{i,j}$ as follows:

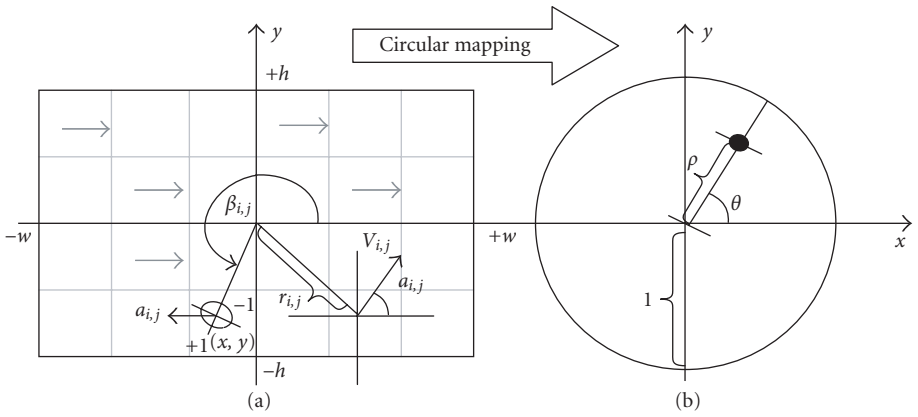

Figure 1: Circular mapping.

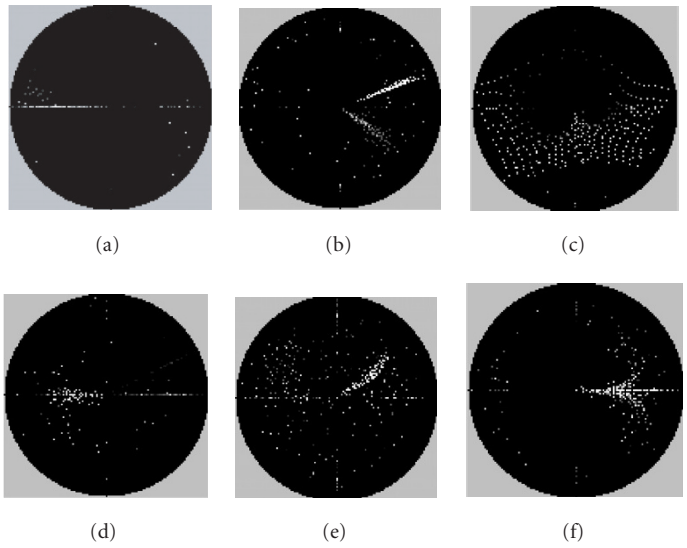

Figure 2: *Examples of EUC patterns.* (a) Camera panning right. There is a light line on the left. (b) Camera tracking. There are two light strips. The one extending to the brim of the EUC results from the camera motion and the other with shorter length results from the object motion. (c) Camera zooming. It presents a special pattern. (d) Irregular object motion with camera being static or moving slightly. (e) Object motion with specific pattern, moving along the orientation of $\pi/4$. (f) A special case, a cube turning around. Its pattern is very distinctive.

$$\mathrm{En}_{i,j} = \sqrt{\Delta x_{i,j}^2 + \Delta y_{i,j}^2}. \tag{1}$$

Since the patterns in original MVF are not sufficiently salient, we map the energy in MVF to a unit circle at first. As shown in Figure 1, we construct rectangular coordinates at the center of the MVF and polar coordinates at the center of the unit circle. The width and height of the MVF are $2w$ and $2h$, respectively. Let $x_{i,j}$ and $y_{i,j}$ denote the position of macro block $\mathrm{MB}_{i,j}$ in rectangular coordinates, then the process of mapping the energy in a MVF to a unit circle can be described as

$$g(\rho,\theta) = \sum_{x_{i,j}=-w}^{+w} \sum_{y_{i,j}=-h}^{+h} \mathrm{En}_{i,j} \quad \text{if } \rho = \bar{r}_{i,j},\ \theta = \alpha_{i,j}, \tag{2}$$

where $g(\rho,\theta)$ is the energy distribution function of the unit circle, $\bar{r}_{i,j} = \sqrt{x_{i,j}^2 + y_{i,j}^2}/\sqrt{w^2+h^2}$ is the normalized distance from macro block $\mathrm{MB}_{i,j}$ to the center of MVF, and $\alpha_{i,j} \in [0, 2\pi]$ is the orientation of motion vector $V_{i,j}$. We call this mapping process *circular mapping* and call the mapped unit circle *energy unit circle* (EUC). In EUC, both object motion and camera motion present distinctive patterns, respectively. Figure 2 gives some examples of EUC patterns of different motions.

3.2. Directional slicing

In order to capture the temporal pattern of motion during a period of time, we extract slices from consecutive EUCs along the temporal axis. As shown in Figure 3, we first divide EUC into n ($n = 4$ in this paper) equiangular opposite

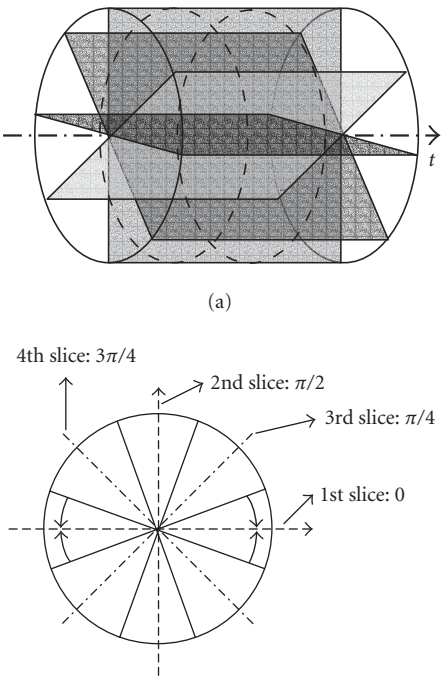

Figure 3: Directional slicing.

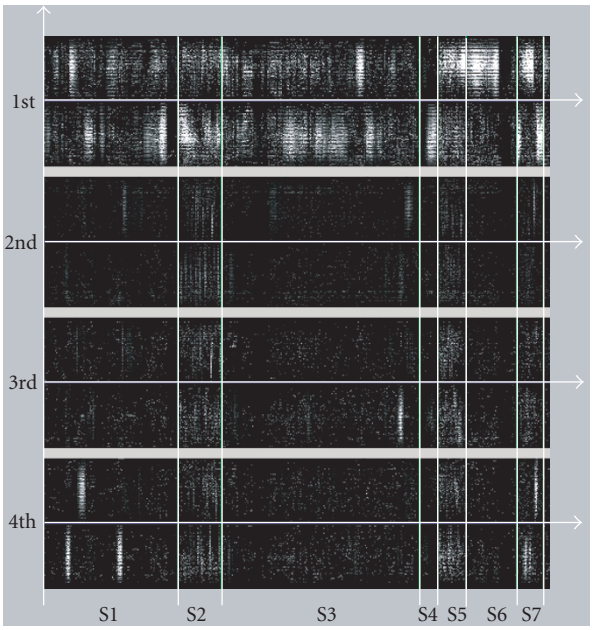

Figure 4: *Directional slices samples from a segment of a basketball game video*. S*n* denotes the shots in video sequence. S1: a bout of offence with a shoot occurred (right court); S2, S5, S7: the camera is tracking a player. The energy distributes evenly among the 4-slices due to the irregular object motion; S3: a bout of offence with a shoot occurred (left court); S4: a specific wipe; S6: a bout of offence without shoot.

sectors with the central lines at 0, $\pi/4$, $\pi/2$, and $3\pi/4$. Then, the energy in each sector is accumulated to the central lines along homocentric circumference. Finally, we extract directional slices from EUC volume at those central lines. This process is called directional slicing.

Figure 4 shows some samples of directional slices, in which the horizontal coordinate is temporal axis and the vertical coordinate indicates the distance from macro block to the center of the MVF. The positive and negative values of vertical coordinate denote the two opposite directions, respectively. Thus, 4-directional slices are able to describe the patterns in 8 directions. In this way, motion intensity, dominant direction, and motion pattern all can be presented with a few gray-level images as the spatial and temporal distribution of energy.

3.3. Moments measuring

The operations described above unveil the motion patterns from the MVF. However, we still need to find an effective method to measure these slice images to generate a compact and quantitative representation. According to Hu's *uniqueness theorem* [16], if a function $f(x, y)$ is piecewise continuous and has nonzero values only in the finite region of the (x, y) plane, then the moments of all orders exist. It can be shown that the moment set $\{m_{pq}\}$ is uniquely determined by $f(x, y)$ and conversely, $f(x, y)$ is uniquely determined by $\{m_{pq}\}$. Therefore, if we describe the directional slices with energy density functions $f_n(x, y)$, the moments can be employed to represent these slices. Since the directional slice images $f_n(x, y)$ have finite area and, in the worst case, are piecewise continuous, moments of all orders exist and a moment set will uniquely describe the information contained in them. In this paper, we select a subset of moments from the zeroth to the fourth order to characterize slices. Assuming that the slice images have the size of $M \times N$, the moments are calculated as

$$m_{pq} = \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} x^p y^q f(x, y), \tag{3}$$

where $(p, q) = \{(0, 0), (1, 0), (0, 1), (2, 0), (0, 2), (3, 0), (0, 3), (4, 0), (0, 4)\}$. Based on these moments, we compute 9 values with specific physical meaning (4), (5), (6), (7), (8), (9), and (10). These values can be normalized values by the size of slices if it is necessary.

The center of mass is computed by (4) and normalized by (5)

$$\mathrm{COM}_x = \frac{m_{10}}{m_{00}}, \qquad \mathrm{COM}_y = \frac{m_{01}}{m_{00}}, \tag{4}$$

$$\overline{\mathrm{COM}_x} = \frac{\mathrm{COM}_x}{M}, \qquad \overline{\mathrm{COM}_y} = \frac{\mathrm{COM}_y}{N} \tag{5}$$

and the radii of gyration is computed by (6), normalized by (7)

$$\mathrm{ROG}_x = \sqrt{\frac{m_{20}}{m_{00}}}, \qquad \mathrm{ROG}_y = \sqrt{\frac{m_{02}}{m_{00}}}, \tag{6}$$

$$\overline{\mathrm{ROG}_x} = \frac{\mathrm{ROG}_x}{M}, \qquad \overline{\mathrm{ROG}_y} = \frac{\mathrm{ROG}_y}{N}. \tag{7}$$

In order to compute the skewness and the kurtosis, the central moment μ_{pq} is required as follows:

$$\mu_{pq} = \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} (x - \mathrm{COM}_x)^p (y - \mathrm{COM}_y)^q f(x, y), \quad (8)$$

where $(p, q) = \{(2, 0), (0, 2), (3, 0), (0, 3), (4, 0), (0, 4)\}$. Then, the skewness and the kurtosis can be obtained, respectively, as

$$Sk_x = \frac{\mu_{30}}{\mu_{20}^{3/2}}, \qquad Sk_y = \frac{\mu_{03}}{\mu_{02}^{3/2}}, \quad (9)$$

$$K_x = \frac{\mu_{40}}{\mu_{20}^2} - 3, \qquad K_y = \frac{\mu_{04}}{\mu_{02}^2} - 3. \quad (10)$$

With the above measures, we define a 9-dimensional feature vector for each directional slice: $Fn = \{m_{00}, \overline{\mathrm{COM}_x}, \overline{\mathrm{COM}_y}, \overline{\mathrm{ROG}_x}, \overline{\mathrm{ROG}_y}, Sk_x, Sk_y, K_x, K_y\}$, where $n \in [0, 3]$. So the total dimension of the feature vector is 36.

3.4. Signed energy

In order to characterize the motion's convergence or divergence relative to the focus of expansion (FOE), a *signed energy* is defined for each macro block $\mathrm{MB}_{i,j}$ as a supplement to the definition in Section 3.3,

$$\mathrm{SEn}_{i,j} = \begin{cases} 1, & \left(|\alpha_{i,j} - \beta_{i,j}| < \dfrac{\pi}{2}\right), \\ -1, & \text{otherwise}, \end{cases} \quad (11)$$

where $\alpha_{i,j}$ still denotes the orientation of motion vector $V_{i,j}$ and $\beta_{i,j}$ is the direction angle of the macro block $\mathrm{MB}_{i,j}$ in MVF, see Figure 1a. The signed energy is also transformed to the directional slices by circular mapping and directional slicing. Then we compute the average signed energy in each directional slice by (12) as follows:

$$\overline{\mathrm{SEn}} = \frac{1}{M \times N} \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} \mathrm{SEn}_s. \quad (12)$$

3.5. Motion texture

With nine energy-distribution measures and one additional signed-energy measure of each directional slice, a $10 \times D$ dimensions vector: $T = \{T^0, T^1, T^n, \ldots, T^{D-1}\}$ is obtained, where $T^n = \{{m_{00}}^n, \overline{\mathrm{COM}_x}^n, \overline{\mathrm{COM}_y}^n, \overline{\mathrm{ROG}_x}^n, \overline{\mathrm{ROG}_y}^n, {Sk_x}^n, {Sk_y}^n, {K_x}^n, {K_y}^n, \overline{\mathrm{SEn}}^n\}$, $n \in [0, D-1]$, and D is the number of slices cut from EUCs. We name this vector *motion texture*, by which all of the motion characteristics in a video clip are extracted effectively and represented compactly. Since $D = 4$ in this paper, we obtain a 40-dimensional vector as our motion description.

4. SEMANTIC VIDEO CLASSIFICATION BASED ON MOTION

Since motion is an important cue to perceive video content and the motion patterns often convey some semantic information, video can be classified into semantic categories based on motion patterns. On the other hand, motion texture is a motion pattern descriptor based on a MVF, which is also a low-level feature of the video. Therefore, we need to find an effective way to map motion texture to semantic conceptions. Considering the complexity of the motion in video and the high dimension of motion texture, we employ kernel SVMs to devise a set of multicategory classifiers to meet this requirement because it has high generalization performance in high-dimensional feature space.

4.1. Kernel SVMs

It is known that SVMs can give an optimal separating hyperplane with a maximal margin if the data is linearly separable. In linearly nonseparable but nonlinearly separable case, the data will be mapped into a high-dimensional space where the two classes of data are more readily separable. Such mapping is formed by a kernel representation of data.

Consider the problem of separating a set of training examples belonging to two classes: $(x_i; y_i)_{1 \le i \le N}$, where each example $x_i \in R^d$, d being the dimension of the input space, belongs to a class labeled by $y_i \in \{-1, +1\}$. Once a kernel $K(\mathbf{x}_i, \mathbf{x}_j)$ satisfying Mercer's condition [17] has been chosen, an optimal separating hyperplane will be constructed in the mapping space. The optimization problem can be achieved by the maximization of the objective function (13) with Lagrange multipliers

$$L(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j), \quad (13)$$

where α_i is Lagrange multipliers. Then, the decision function will be

$$f(\mathbf{x}) = \mathrm{sgn}\left(\sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right). \quad (14)$$

Possible choices of kernel functions include polynomial, Gaussian radial-basis function (RBF) and multilayer-perception function. In this paper, we use Gaussian RBF kernel, which is defined in (15), since it was empirically observed to perform better than the other two

$$K_{\mathrm{Gaussian}}(\mathbf{x}, \mathbf{y}) = \exp(-\rho \|\mathbf{x} - \mathbf{y}\|^2). \quad (15)$$

In this case, the number of centers or the number of support vectors, the centers themselves or the support vectors, the weights (α_i), and the threshold (b) are all produced automatically by the SVMs training and give excellent results.

4.2. Multiclass classification

SVMs are designed for binary classification. When we want to discriminate several classes simultaneously, there are three solutions to construct a multiclass classifier: (1) to modify the design of the SVMs; (2) to combine binary classifiers by one-against-one applying pairwise comparisons between classes; (3) to combine binary classifiers by one-against-others

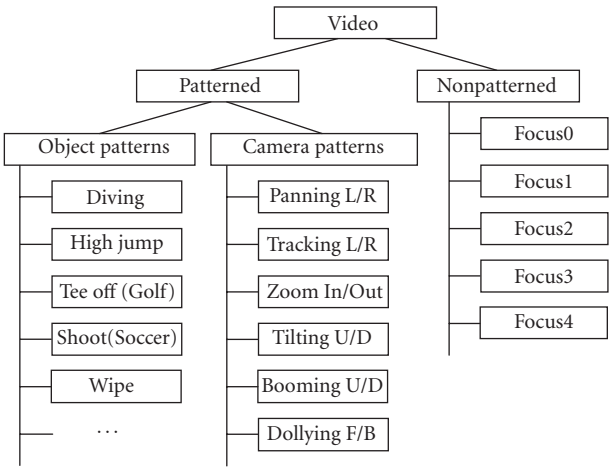

Figure 5: Classification scheme.

comparing a given class with all the others put together. It was proved that the accuracies of these methods are almost the same in [13]. Therefore, we choose the third method because it has the lowest complexity.

In a one-against-others solution, n-hyperplanes are constructed, where n is the number of classes. Each hyperplane separates one class from the other classes. Namely, we get n decision functions $(f_k)_{1\le k\le n}$. The class of a new point $\mathbf{x}$ is given by

$$C(\mathbf{x}) = \arg\max_k \{f_k(\mathbf{x})\}, \tag{16}$$

that is, the class with the largest decision function.

4.3. Motion pattern-based classification scheme

Since video clips do not always have salient motion patterns and the semantic conceptions are not always conveyed by motion patterns, not all of the video clips can be mapped to semantic conception based on motion patterns. Therefore, we define a classification scheme, as shown in Figure 5, to facilitate semantic video classification.

At fist, video clips are classified into two basic classes: patterned and nonpatterned. Then, the patterned class is further classified into two classes: object patterns and camera patterns. In most of the cases, object motion cannot be discriminated from camera motion clearly in video. So we use the following rules to define the two categories. If object motion is much more dominant than camera motion in a video clip, this clip will be considered as the case of object patterns although sometimes there are also some slight camera motions. On the other hand, we only classify the standard camera operations into camera patterns in which the camera motion is so dominant that object motion can be ignored, such as panning, tracking, zooming, and so on.

In the case of object patterns, semantic conceptions are conveyed by object motions and camera motions together. For example, a photographer usually lets the camera track a ball's trail after focusing on the players' motions when a shot occurs in a basketball game. Since the semantic conceptions are not countable in real world, the number of subclasses is also extendable in our scheme. They can be defined by users according to semantic conceptions or events.

If the camera focuses on the objects moving irregularly and both camera and objects do not have dominant motion, then there will be no salient motion patterns. We categorize such clips as nonpatterned. In this case, we only rank the intensity of motion by 5 levels like motion descriptors defined in MPEG-7 [18]. The 5-level subcategories are labeled from Focus0 to Focus4 in our scheme. The video clips in class Focus0 are static with the motion energy near zero, while the ones in class Focus4 have the highest motion intensity.

Within such classification scheme, we can classify all types of video clips into subcategories by a multiclass classifier. In order to improve the speed and accuracy of classification, the classification of video clips can be in 3 steps: (1) to discriminate patterned from nonpatterned by a binary classifier; (2) to discriminate object patterns from camera patterns by another binary classifier; and (3) to classify all of the video clips into subcategories within each basic category by a multiclass classifier, respectively. In this way, we need three multiclass classifiers.

4.4. Experiments

We have build up a video shot database of 10 hours with real-world video programs including science and educational films, sightseeing videos, stage performances, and sports videos. These videos were firstly segmented into shots. Then, motion texture was extracted from each shot as the motion pattern descriptor. In our experiments, SVMTorch [19] is used to train the models and construct classifiers, in which the RBF kernel is selected.

During the experiments, the shots in the database are manually classified into different subcategories according to our classification scheme first. From them, two nonpatterned classes, two camera patterns classes, and eight object patterns classes are chosen as the test set which have sufficient samples. Then, we train the models for each subcategory with about half the samples and test them with the other half of the samples. In addition, two binary classifiers, patterned/nonpatterned and object pattern/camera pattern are also trained and tested. The experimental results are listed in Table 1.

From Table 1, we can see that (1) the proposed method is very effective for the clips belonging to patterned classes; (2) the performance of nonpatterned classes is slightly poorer because of the lack of salient patterns; (3) both multiclass classifiers and two binary classifiers can achieve high classification accuracy. The average accuracy of our classification results approaches 94%.

In addition, we have designed a comparison experiment with 4 solutions as following: (A) SVMs + motion texture proposed in this paper; (B) SVMs + conventional motion features; (C) KNN + motion texture; and (D) KNN + conventional motion features. The conventional motion features include the orientation of dominant motion and the average

Table 1: Classification results.

Model[a]	Class	NoTTS	NoSVs	Acc.
OP-1	Diving or not	266/260	83	96.19%
OP-2	High jump or not	261/255	65	95.81%
OP-3	Race or not	258/260	90	91.95%
OP-4	Tee off (Golf) or not	200/210	57	96.70%
OP-5	Shoot (Soccer) or not	255/275	91	90.98%
OP-6	Team offense/defense (Basketball) or not	246/250	87	93.47%
OP-7	Penalty shot (Basketball) or not	218/220	49	95.29%
OP-8	Wipe or not	180/150	38	98.14%
CP-1	Tracking left or not	260/268	36	97.08%
CP-2	Zoom-out or not	198/180	42	96.77%
NP-1	Fcous0 or not	280/292	97	88.32%
NP-2	Focus4 or not	260/275	99	84.87%
BC-1	With pattern or not	300/300	75	91.78%
BC-2	Camera pattern or not	300/300	79	92.64%
Avg.	—	—	—	93.57%

[a]OP: object pattern; CP: camera pattern; NP: nonpattern; BC: binary classifier; NoTT: number of training/test samples; NoSVs: number of support vectors.

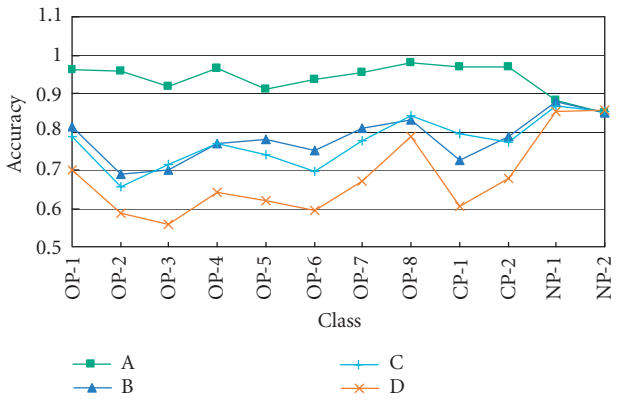

Figure 6: Accuracy curves.

magnitude of motion vectors. We test all of multiclass classifiers with the same test set. The accuracy curves are drawn in Figure 6.

Observing the accuracy curves in Figure 6, we can draw the following conclusions: (1) solution A always outperforms other solutions in the case of patterned classes; (2) the curves of solutions B and C are almost the same, both are slightly better than that of solution D; (3) the curve of solution A is smoother than others, indicating that the proposed method is the most stable; (4) for those clips belonging to nonpatterned classes, the accuracy values of four solutions are very close. It is because if there is not any salient motion pattern, *motion texture* only captures the motion direction and intensity like the conventional statistical methods.

5. MOTION-BASED SHOT RETRIEVAL IMPROVEMENT

Motion texture, as a compact motion pattern descriptor, can be used directly in motion-based shot retrieval. Moreover, as a motion feature, motion texture can also be combined with other visual features for more complex retrieval. In this section, we first apply motion texture in motion-based video retrieval. Experimental results indicate that it outperforms other existing motion feature representations. In addition, we take advantage of the classification results, described in Section 4, to further improve traditional retrieval approach. The performance is improved significantly.

5.1. *Motion-based shot retrieval*

To apply a content descriptor in video retrieval, we first need to define the similarity based, upon which video shots are ranked against a query. In this section, we define a similarity measure for motion texture. Since the dynamic range of each component of motion texture is quite different, the normalization is indispensable when we compare two motion texture vectors. Assuming that we have a video-shot database, the motion texture is extracted from each shot. Then we normalize each component of vectors by the inverse of the standard variance. The standard variance of kth component is

$$\sigma_k = \sqrt{\frac{\sum_{l=1}^{L} \left(v_k^{(l)} - \bar{v}_k\right)^2}{L}}, \tag{17}$$

where $v_k^{(l)}$ denotes the kth component of the lth feature vector, L is the number of samples in the database, and $\bar{v}_k$ is the

mean of $v_k^{(l)}$ which is computed as

$$\bar{v}_k = \frac{\sum_{l=1}^{L} v_i^{(l)}}{L}. \quad (18)$$

Using the normalization coefficients as weights, we adopt weighted Euclidean distance to measure the similarity of motion texture. When comparing two motion texture vectors T^a and T^b in a video-clip database, the similarity is defined as

$$\mathrm{Sim}\left(T^a, T^b\right) = \sqrt{\sum_{k=1}^{n\times S} \frac{1}{\sigma_k^2}\left(v_k^a - v_k^b\right)^2}. \quad (19)$$

In this way, the modality-dependent amplitude difference is reduced effectively. The effectiveness of this similarity measurement has been verified by our experiments.

5.2. Classification-based shot retrieval

The traditional content-based retrieval methods usually only depend on a low-level feature. So the description capability of a low-level feature is a key fact in retrieval performance. For a specific problem, one or a set of low-level features perhaps are effective. However, the retrieval power of any low-level feature, when applied to video databases of general content, is very limited. In this section, we take advantage of the semantic classification results to further improve content-based retrieval method.

Generally, if the category of query sample has been known, we can rank the samples within the same category of a query sample higher than the other samples in the database. Thus, the accuracy of retrieval will be improved in this way. But it is required that all samples in the database as well as the query sample are labeled correctly. In fact, there always exist some samples misclassified with any automatic method. If the samples in the database are misclassified, the performance is not affected severely; whereas if a query sample is misclassified, the retrieval results can be totally wrong. Therefore, we should reduce the misclassification risk of a query sample and this is done by merging the retrieval results from different categories in our retrieval scheme.

It is a feasible method to weight the retrieval results based on categories. If the probabilities of a query belonging to each category are obtained, these probability values can be used as weights of similarity between the query and the samples from different categories. Since the standard SVMs only provide one or a set of decision functions, we cannot obtain a calibrated *posterior* probability from the result of classification directly. J. Platt proposed a method to extract probabilities from SVMs outputs [20]. We adopt this approach to weight the similarity from different categories. Then, the classification-based similarity measure $\mathrm{CSim}(EFn^q, EFn_c^{(l)})$ is defined as

$$\mathrm{CSim}\left(EFn^q, EFn^{(l)}\right) = \mathrm{Sim}\left(EFn^q, EFn_c^{(l)}\right) \times p(c \mid q), \quad (20)$$

where $\mathrm{Sim}(EFn^q, EFn_c^{(l)})$ is computed as in (19). The only difference is that $EFn_c^{(l)}$ has been labeled as class c, $p(c \mid q)$ denotes the probability of query q belonging to the class c. This probability is obtained by the method proposed in [20]. The experimental results indicating this method is effective for the rank merging.

5.3. Experiments

We have evaluated the proposed representation and similarity measure on a motion-based shot retrieval system with the same video database as Section 3. There are about 10000 shots in this video database, which are segmented from about 10 hours' real-world videos. For comparison purpose, we implemented three approaches on the same test data. The first one is a conventional method based on motion intensity and dominant directions. The second one adopts motion texture as motion descriptor. The last one is classification-based shot retrieval, in which the semantic classification scheme proposed in Section 4 is employed with the similarity modifications presented above. From each shot, the motion features used in the three methods are extracted. According to the similarity of motion patterns, we manually classify the shots in the video database into 54 classes. Then, 14 classes are selected from such 54 classes as a test set. The test set includes 4 classes without salient motion, 8 classes with an object motion pattern, and 2 classes with a camera motion pattern. They are marked with NP, OP, and CP, respectively, in Table 2. In each class, each member is picked out as a query sample in turn and the rest members are used as ground truth. The performance is evaluated by *average normalized modified retrieval rank* (ANMRR) (which is the smaller, the better) and *average retrieval recall* (ARR) (which is the larger, the better), as proposed in MPEG-7 [21]. Their definitions can be found in the appendix. The experimental results are listed in Table 2.

Table 2 shows that the motion texture-based method always outperforms the conventional method, especially for the shots with salient object motion patterns. Since the camera motions are very distinctive, they can be easily identified by both methods. In the case of shots without salient patterns, the improvement by the proposed method is limited because only motion intensity and dominant direction are contributing. The best performance is obtained by a classification-based method. Its ANMRR approaches 0.15, and ARR is above 0.9. This observation concludes that the proposed motion descriptor is effective for motion-based video classification and retrieval, and the proposed classification probability-based video retrieval scheme is effective.

6. CONCLUDING REMARKS

In this paper, we have presented a generic motion representation, named *motion texture*. Most of the major motion characters are preserved within this representation. Based on such a representation, we not only are able to effectively improve the performance of motion-based video retrieval, but also have devised a semantic classification scheme by which the motion patterns can be mapped to semantic conceptions. Experimental results indicate that motion texture is a compact, generic, and effective representation of a motion

Table 2: Performance evaluation.

Query Shot	Conventional method		Motion texture-based method		Classification-based method	
	ANMRR	*ARR*	*ANMRR*	*ARR*	*ANMRR*	*ARR*
NP-1	0.6765	0.4998	0.5333	0.6065	0.3998	0.7806
NP-2	0.6603	0.4982	0.5011	0.6754	0.3871	0.7962
NP-3	0.6499	0.4371	0.5027	0.6508	0.4002	0.7533
NP-4	0.6598	0.4009	0.5249	0.6192	0.3906	0.7455
OP-1	0.4867	0.5865	0.2325	0.8154	0.1056	0.8961
OP-2	0.3944	0.7011	0.1305	0.8745	0.0812	0.9652
OP-3	0.3293	0.6581	0.0992	0.9046	0.0687	0.9711
OP-4	0.3943	0.6402	0.1375	0.8959	0.0803	0.9667
OP-5	0.4149	0.6749	0.2043	0.8076	0.0753	0.8992
OP-6	0.5061	0.6574	0.1214	0.8832	0.0599	0.9581
OP-7	0.2012	0.7625	0.0000	1.0000	0.0100	1.0000
OP-8	0.2835	0.8081	0.0000	1.0000	0.0100	1.0000
CP-1	0.1628	0.7963	0.0825	1.0000	0.0250	1.0000
CP-2	0.2037	0.8755	0.0933	1.0000	0.0311	0.9900
Avg.	0.4302	0.6426	0.2259	0.8380	0.1518	0.9087

pattern. When we apply the semantic classification results in a video retrieval process, the retrieval performance is further improved significantly. The proposed motion descriptor and the classification scheme provide a solution to remove the two barriers in content-based video retrieval: the lack of efficient content representation and effective method of bridging the gap between low-level features and semantic conceptions.

The proposed methods can be further improved by the more efficient measure of directional slices, the more effective merging method for classification-based retrieval, and the more accurate motion estimation. Besides the content-based classification and retrieval, the motion texture can also be used to solve other motion-related problems, such as event detection in surveillance. To extend the proposed framework in these directions is our future work.

APPENDIX

In the retrieval experiments, we adopt the ANMRR and the ARR, recommended by MPEG-7 core experiments document [21], as the evaluation criteria. Given a query set and the corresponding ground truth data, the ANMRR and ARR values all range between [0,1]. A low ANMRR value denotes a high retrieval rate with relevant items ranked at the top. Compared with ANMRR, a high ARR value indicates a high retrieval rate.

Let the number of ground truth shots for query q be $\mathrm{NG}(q)$ and $k = \min(4 \times \mathrm{NG}(q), 2 \times GTM)$, where GTM is $\max(\mathrm{NG}(q))$ for all queries. For each ground truth shot k retrieved in the top K retrievals, compute the rank of the shot, $\mathrm{Rank}(k)$. Counting the rank of the first retrieved item as 1 and the rank of $(K+1)$ is assigned to those ground truth shots not in the top K retrievals. The *modified retrieval rank* ($\mathrm{MRR}(q)$) is computed by (A.1)

$$\mathrm{MRR}(q) = \sum_{k=1}^{\mathrm{NG}(q)} \frac{\mathrm{Rank}(k)}{\mathrm{NG}(q)} - \frac{1+\mathrm{NG}(q)}{2}. \tag{A.1}$$

With (A.1), the *normalized modified retrieval rank* (NMRR) is defined as (A.2)

$$\mathrm{NMRR}(q) = \frac{\mathrm{MRR}(q)}{K - \mathrm{NG}(q)/2 + 0.5}, \tag{A.2}$$

where the value of NMRR is in the range of $[0, 1]$. Finally, the average NMRR of all values is computed over all queries to yield the ANMRR.

With the same assumption, the *retrieval recall* (RR) is defined as

$$RR(q) = \sum_{k=1}^{\mathrm{NG}(q)} \frac{\mathrm{rank}(k)}{\mathrm{NG}(q)}, \tag{A.3}$$

and the ARR is computed over all queries the same as ANMRR.

REFERENCES

[1] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performances of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.

[2] E. Ardizzone and M. La Cascia, "Video indexing using optical flow field," in *Proc. IEEE International Conference on Image Processing*, vol. 3, pp. 831–834, Lausanne, Switzerland, September 1996.

[3] D. Zhong and S.-F. Chang, "AMOS: an active system for MPEG-4 video object segmentation," in *Proc. IEEE International Conference on Image Processing*, vol. 2, pp. 647–651, Chicago, Ill, USA, October 1998.
[4] H. S. Sawhney and S. Ayer, "Compact representations of videos through dominant and multiple motion estimation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 814–830, 1996.
[5] S.-F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong, "VideoQ: An automated content based video search system using visual cues," in *Proc. ACM Multimedia*, pp. 313–324, Seattle, Wash, USA, November 1997.
[6] S. L. Peng, "Temporal slice analysis of image sequences," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 283–288, Maui, Hawaii, USA, June 1991.
[7] F. Liu and R. W. Picard, "Finding periodicity in space and time," in *Proc. IEEE International Conf. on Computer Vision*, pp. 376–383, Bombay, India, January 1998.
[8] C. W. Ngo, T. C. Pong, H. J. Zhang, and R. T. Chin, "Motion characterization by temporal slice analysis," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 768–773, Hilton Head Island, SC, USA, June 2000.
[9] A. M. Dawood and M. Ghanbari, "Scene content classification from MPEG coded bit stream," in *Proc. 1999 IEEE 3rd Workshop on Multimedia Signal Processing*, pp. 253–258, Copenhagen, Denmark, September 1999.
[10] J. Huang, Z. Liu, Y. Wang, Y. Chen, and E. K. Wong, "Integration of multimodal features for video classification based on HMM," in *Proc. 1999 IEEE 3rd Workshop on Multimedia Signal Processing*, pp. 53–58, Copenhagen, Denmark, September 1999.
[11] W. Zhou, A. Vellaikal, and C. C. Kuo, "Video analysis and classification for MPEG-7 applications," in *Proc. IEEE International Conference on Consumer Electronics*, Los Angeles, Calif, USA, June 2000.
[12] R. K. Rao, K. R. Ramakrishnan, S. H. Srinivas, and N. Balakrishnan, "Neural net based scene change detection for video classification," in *Proc. 1999 IEEE 3rd Workshop on Multimedia Signal Processing*, pp. 247–252, Copenhagen, Denmark, September 1999.
[13] O. Chapelle, P. Haffner, and V. N. Vapnik, "Support vector machines for histogram-based image classification," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1055–1064, 1999.
[14] E. Ardizzone, M. La Casia, and D. Molinelli, "Motion and color-based video indexing and retrieval," in *Proc. IEEE International Conference on Pattern Recognition*, pp. 135–139, Vienna, Austria, August 1996.
[15] E. Ardizzone, M. La Cascia, A. Avanzato, and A. Bruna, "Video indexing using MPEG motion compensation vectors," in *IEEE International Conference on Multimedia Computing and Systems*, vol. 2, pp. 725–729, Florence, Italy, June 1999.
[16] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.
[17] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA, 1998.
[18] MPEG Video Group, *MPEG-7 Visual part of eXperimentation Model (XM) Version 2.0*, ISO/MPEG, December 1999.
[19] R. Collobert and S. Bengio, "SVMTorch: Support vector machines for large-scale regression problems," *Journal of Machine Learning Research*, vol. 1, pp. 143–160, February 2001.
[20] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*, A. J. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, Eds., MIT Press, Cambridge, Mass, USA, 1999.
[21] MPEG Video Group, "Description of core experiments for MPEG-7 color/texture descriptors," ISO/MPEG JTC1/SC29/WG11 MPEG98/M2819, July 1999.

Yu-Fei Ma received his B.S. degree from Harbin Engineering University, China in 1994 and received his M.S. degree in computer science from Tsinghua University, China in 2000. Mr. Ma joined Microsoft Research Asia in 2000 and now is an Associate Researcher of Multimedia Computing Group. His current research interests are in video content analysis, image processing, and pattern recognition. He has published a number of papers in these fields. From 1994 to 1997, Mr. Ma was engaged in computer network system analysis as a System Engineer.

Hong-Jiang Zhang received his B.S. from Zhengzhou University and his Ph.D. from the Technical University of Denmark, China, both in electrical engineering, in 1982 and 1991, respectively. From 1992 to1995, he was with the Institute of Systems Science, National University of Singapore, where he led several projects in video and image content analysis and retrieval and computer vision. He also worked at the MIT Media Lab in 1994 as a visiting Researcher. From 1995 to 1999, he was a Research Manager at Hewlett-Packard Labs, where he was responsible for research and technology transfers in the areas of multimedia management, intelligent image processing, and Internet media. In 1999, he joined Microsoft Research Asia, where he is currently a Senior Researcher and Assistant Managing Director in charge of media computing and information processing research. Dr. Zhang is a member of the ACM and a senior member of the IEEE. He has authored 3 books, over 200 refereed papers and book chapters, 7 special issues of international journals on image and video processing, content-based media retrieval, and computer vision, as well as over 30 patents or pending applications. He currently serves on the editorial boards of five IEEE/ACM journals and a dozen committees of international conferences.